RSAC 2025 Conference

Many Voices.
One Community.

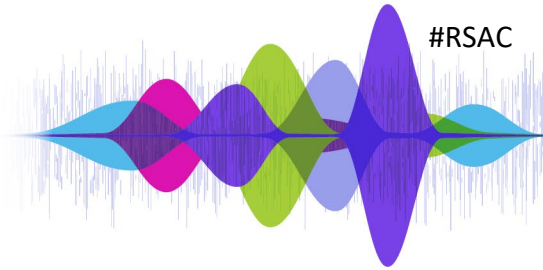
SESSION ID: CRYP-W08A

Secure Computation with Penalties, Revisited: How to Handle Coins with Various Colors

Takeshi Nakai

Assistant Professor
Toyohashi University of Technology
https://sites.google.com/view/takeshi-nakai

Disclaimer



Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the presenters individually and, unless expressly stated to the contrary, are not the opinion or position of RSA Conference LLC or any other co-sponsors. RSA Conference LLC does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented.

Attendees should note that sessions may be audio- or video-recorded and may be published in various media, including print, audio and video formats without further notice. The presentation template and any media capture are subject to copyright protection.

© 2025 RSA Conference LLC or its affiliates. The RSAC and RSAC CONFERENCE logos and other trademarks are proprietary. All rights reserved.

Outline

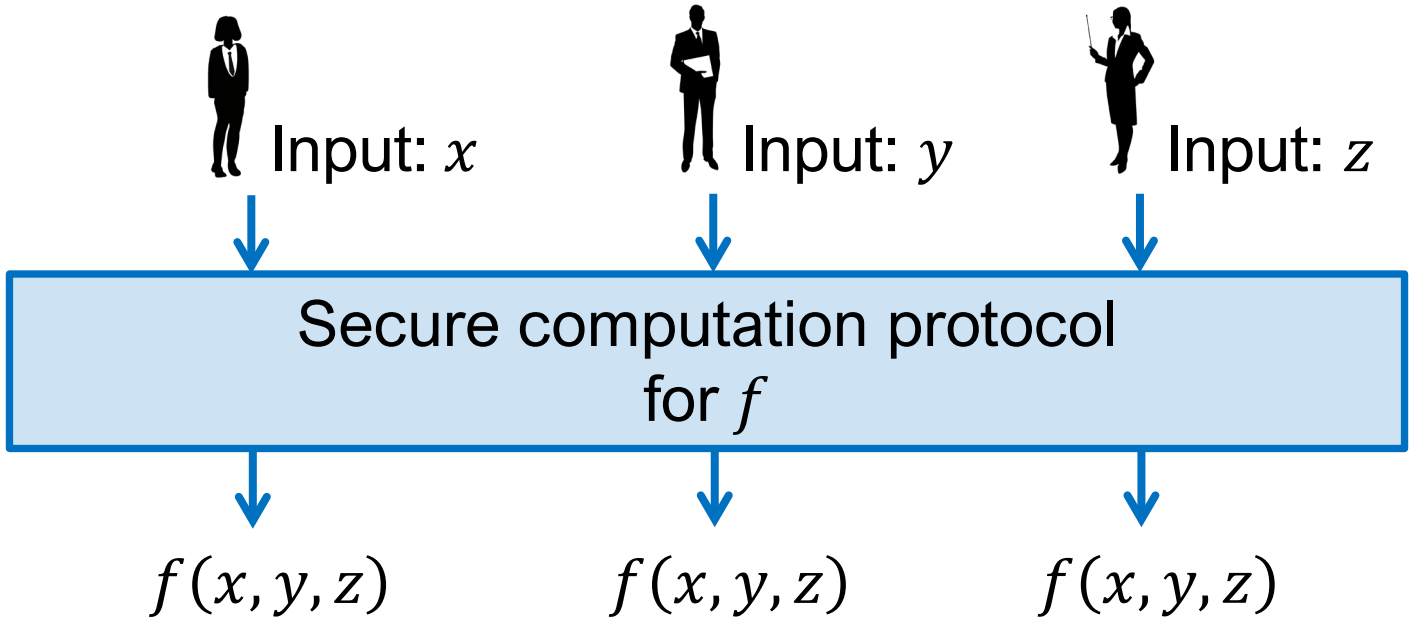
- Background
  - Secure computation with financial transactions
  - Existing frameworks to handle financial transactions in secure computation
    - Public ledger
    - **Security with coins**
- This Work
  - Motivation: Overcome a limitation of security with coins
  - Our Contribution
    - New framework: **security with color coins**
    - Security proof in our framework

Outline

- Background
  - Secure computation with financial transactions
  - Existing frameworks to handle financial transactions in secure computation
    - Public ledger
    - **Security with coins**
- This Work
  - Motivation: Overcome a limitation of security with coins
  - Our Contribution
    - New framework: **security with color coins**
    - Security proof in our framework

Secure Computation

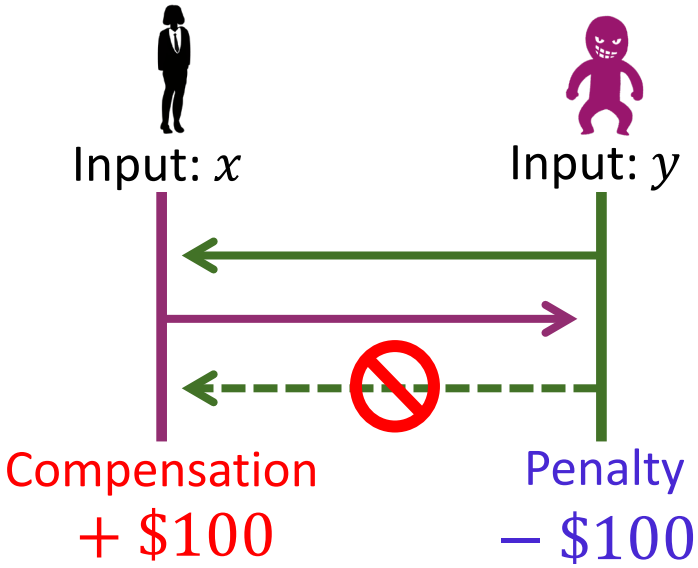
- Setup: Mutually distrustful parties hold private inputs
- Goal : Jointly perform some computation without exposing their inputs
  - even if some parties collude or behave maliciously



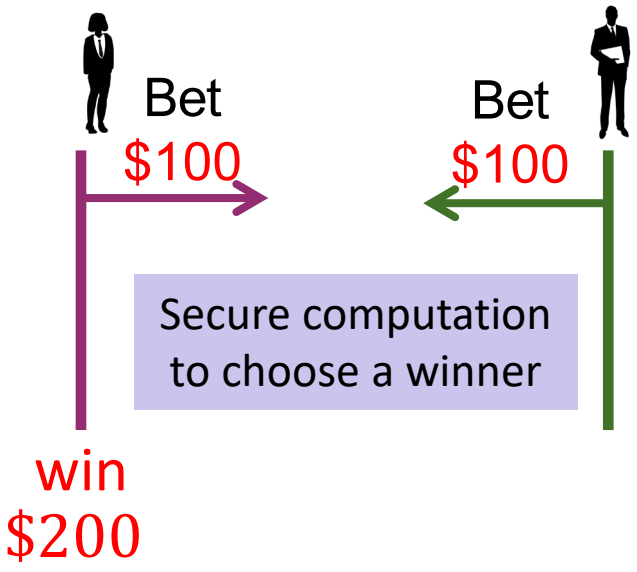
Secure Computation with Financial Transactions

Some secure computation protocols involve financial transactions

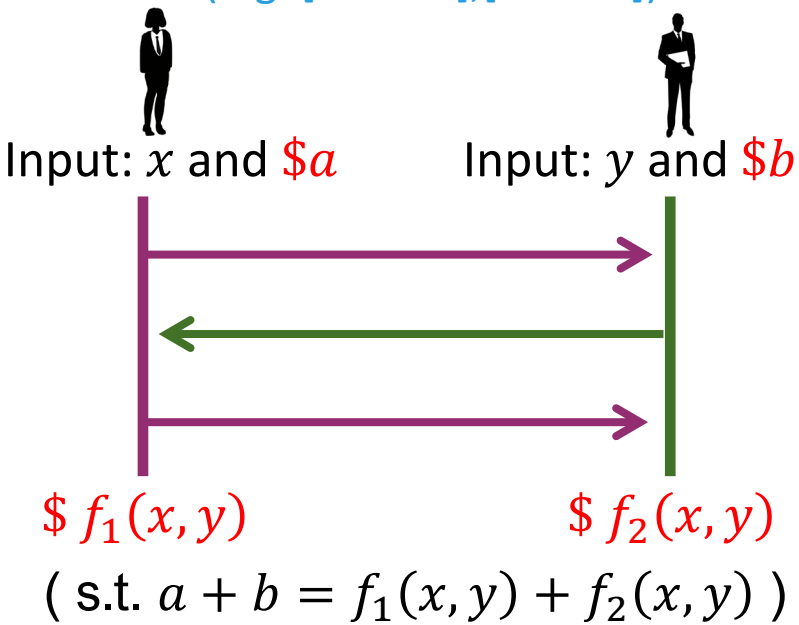
Financial penalties on malicious behaviors
(e.g. [BK14],[FHKS22])



Secure lottery
(e.g. [BK14],[BZ17])



Secure cash distribution
(e.g. [KMB15],[BDD20])



(s.t. $a + b = f_1(x,y) + f_2(x,y)$)

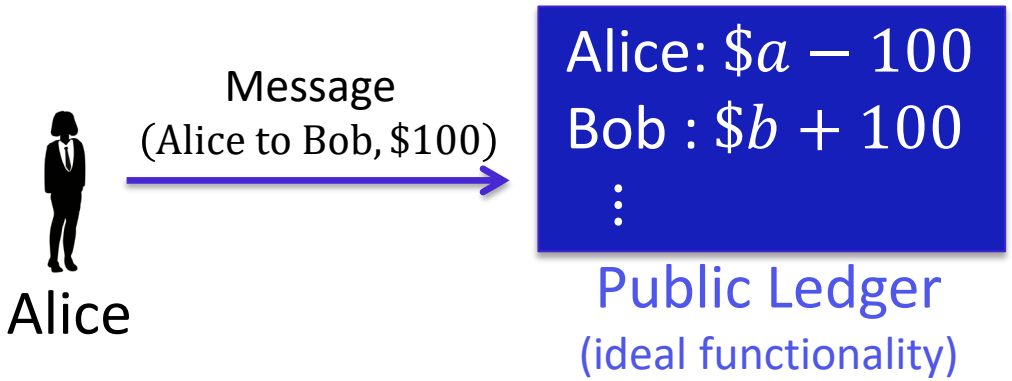
[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.
[FHKS22] S. Faust et al. "Financially Backed Covert Security," PKC2022.
[BZ17] M. Bartoletti and R. Zunino, "Constant-Deposit Multiparty Lotteries on Bitcoin," FC2017.
[KMB15] R. Kumaresan et al., "How to Use Bitcoin to Play Decentralized Poker," CCS2015.
[BDD20] C. Baum et al. "Insured MPC: Efficient Secure Computation with Financial Penalties, FC2020.

Fundamental Problem

How to formalize "money"?

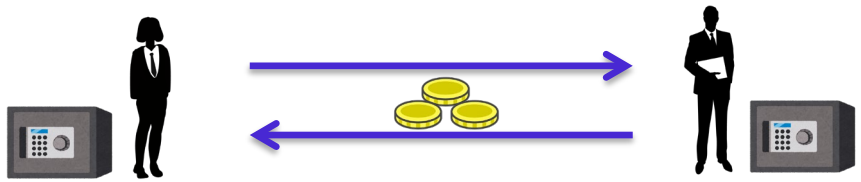
- Existing solutions

Solution 1: Public Ledger e.g. [KZZ16],[DEF18]



- Formulated in the Universal Composition (UC)
- Financial transactions are expressed in the functionality
- Intended to capture blockchain-based currencies

Solution 2: Security with Coins [BK14]



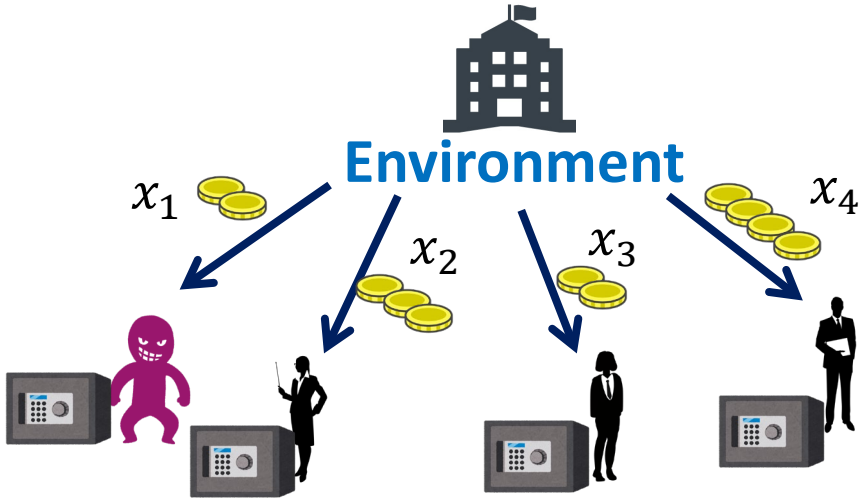
- Perform protocols with special atomic entities called **coins**
- Each party has a special storage to hold coins
- Intended to capture overall currencies
  - ➢ Not only blockchain-based currencies

[KZZ16] A. Kiayias et al., "Fair and Robust Multi-party Computation Using a Global Transaction Ledger," EUROCRYPT2016.
[DEF18] S. Dziembowski et al., "FairSwap: How To Fairly Exchange Digital Goods," CCS2018.
[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

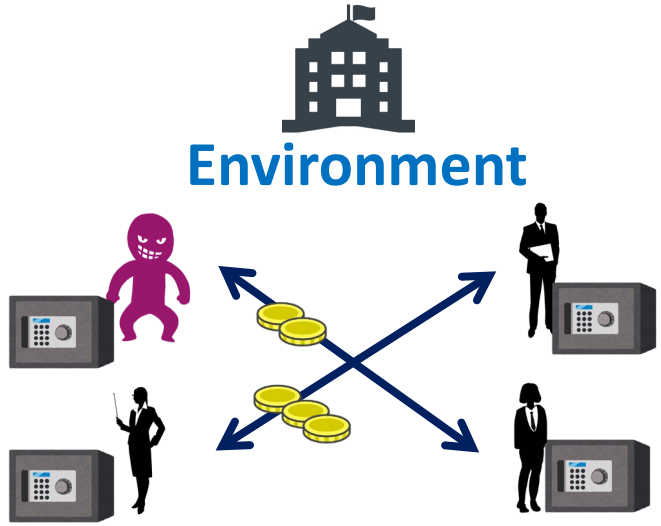
Security with Coins [BK14]

- Based on the UC framework
  - Security is defined as the ideal/real simulation paradigm
    - Environment serves as an interactive distinguisher
- Parties can handle (i.e. hold, send, and receive) coins
  - Only environment has power to create new coins

Ideal-world simulators need to simulate views regarding coins in addition to standard messages



Onset of a protocol execution
(Distribute coins in addition to input values)



During a protocol execution

RSAC 2025 Conference

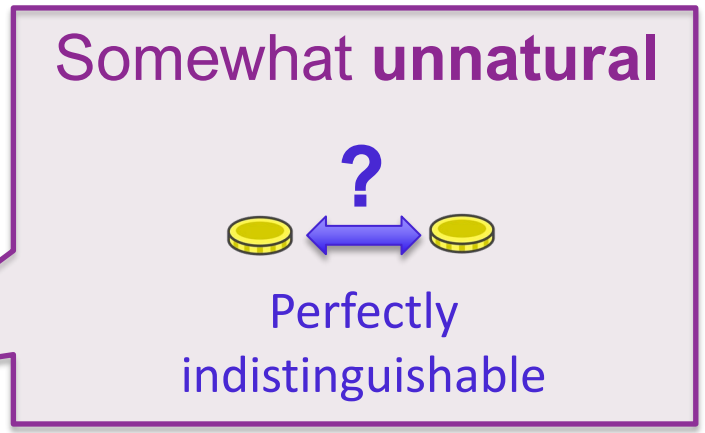
Security with Coins [BK14]

- Based on the UC framework
  - Security is defined as the ideal/real simulation paradigm
    - Environment serves as an interactive distinguisher
- Parties can handle (i.e. hold, send, and receive) coins
  - Only environment has power to create new coins
- Definition of coins (intended to capture overall currencies)
  i. Atomic entities
  ii. Can be transferred between parties
  iii. (Each coin) has a unique owner
  iv. Cannot be duplicated and forged
  v. Indistinguishable from one another

[BK14] I. Bentov and R. Kumaresan, “How to Use Bitcoin to Design Fair Protocols,” CRYPTO2014.

Security with Coins [BK14]

- Based on the UC framework
  - Security is defined as the ideal/real simulation paradigm
    - Environment serves as an interactive distinguisher
- Parties can handle (i.e. hold, send, and receive) coins
  - Only environment has power to create new coins
- Definition of coins (intended to capture overall currencies)
  i. Atomic entities
  ii. Can be transferred between parties
  iii. (Each coin) has a unique owner
  iv. Cannot be duplicated and forged
  v. Indistinguishable from one another



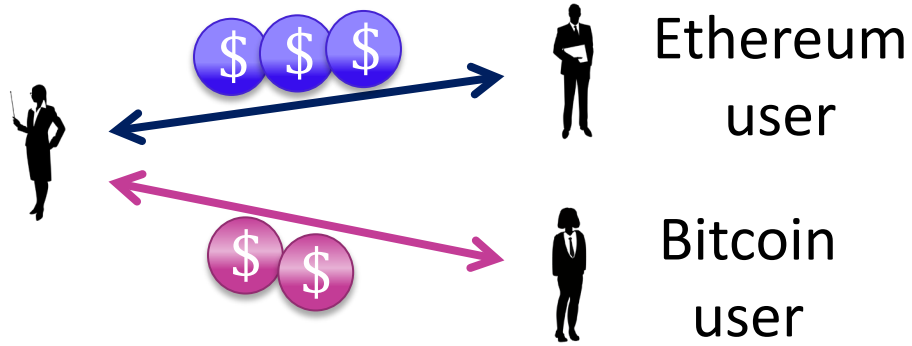
[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

Observation on “Indistinguishable” Coins

- Indistinguishable property of coins
  - Seem to be introduced for security proof (i.e. useful to construct simulators)
  - The framework **CANNOT** deal with monetary entities such that:
    - Distinguishable from each other
    - Have mutually distinct prices

Examples

Coexistence of multiple kinds of currencies



Unique identifier (Distinguishable currency)

Outline

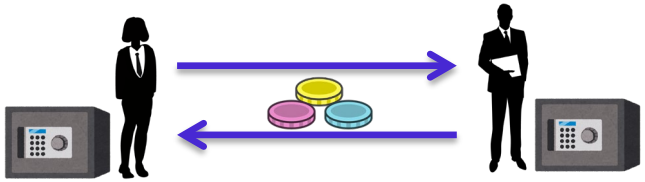
- Background
  - Secure computation with financial transactions
  - Existing frameworks to handle financial transactions in secure computation
    - Public ledger
    - **Security with coins**
- This Work
  - Motivation: Overcome a limitation of security with coins
  - Our Contribution
    - New framework: **security with color coins**
    - Security proof in our framework

Motivation of This Work

- Remove the indistinguishable property from coins to make the framework a more general

- Build a new framework that **CAN** deal with monetary entities such that
  - Distinguishable from each other
  - Have mutually distinct prices

Our Contribution



- Propose a new framework: security with **color coins**
  - Each coin has an identifiable color (i.e. distinguishable from each other)
  - Price can be determined for each color (i.e. mutually distinct prices)

- Realize a protocol for **secure computation with penalties** based on **color coins**
  - Achieve fairness by imposing financial penalties on adversarial parties
  - Almost the same construction as [BK14]'s protocol (based on indistinguishable coins)
  - Security proof without relying on the indistinguishable property
    - By modifying the ideal functionality of secure computation with penalties without loss of the essence, i.e. fairness with penalties

[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

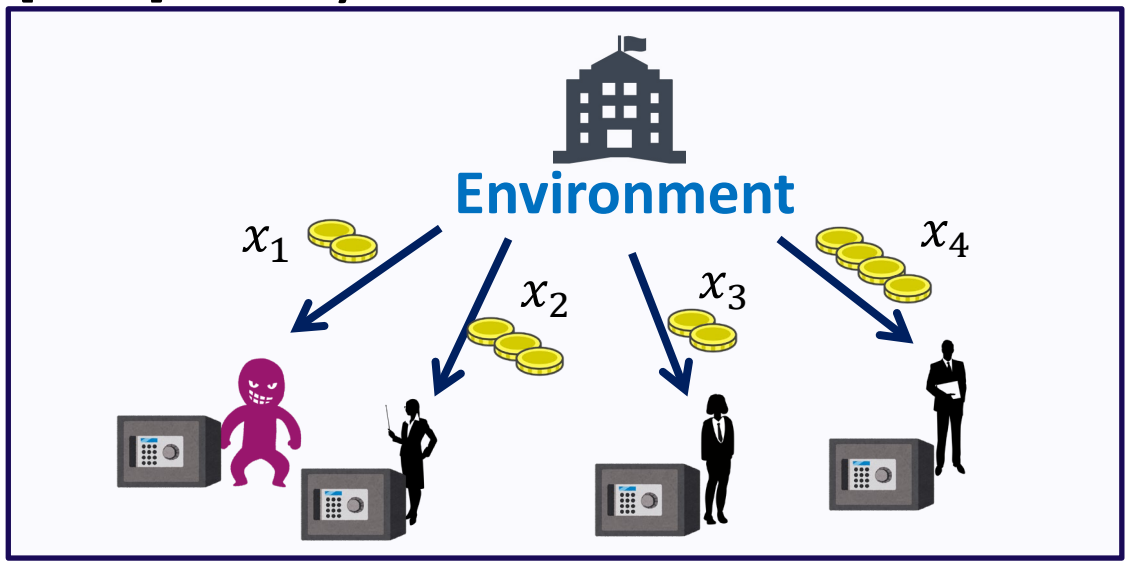
Security with Color Coins

- Based on the UC framework
  - Security is defined as the ideal/real simulation paradigm
- Parties can handle (i.e. hold, send, and receive) color coins
  - Only environment has power to create new color coins
- Definition of **color coins** (intended to capture overall entities that have asset value)
  i. Atomic entities
  ii. Can be transferred between parties
  iii. (Each color coin) has a unique owner
  iv. Cannot be duplicated and forged
  v. Indistinguishable from one another that has the same color
  vi. The colors cannot be changed

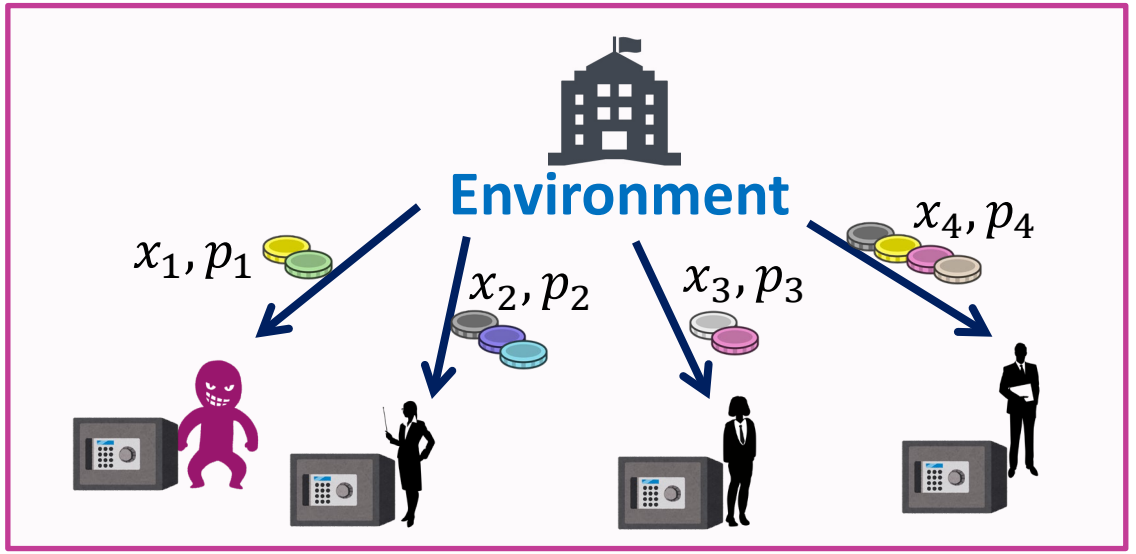
Model Overview

- Security with color coins
  - p_i: price function that determines a price of each color ($\text{colors} \rightarrow \text{positive integers}$)
  - Environment distributes input values, color coins, and **price functions** at the onset
    - Price function is for each party

[BK14] Security with Coins



[This work] Security with Color Coins

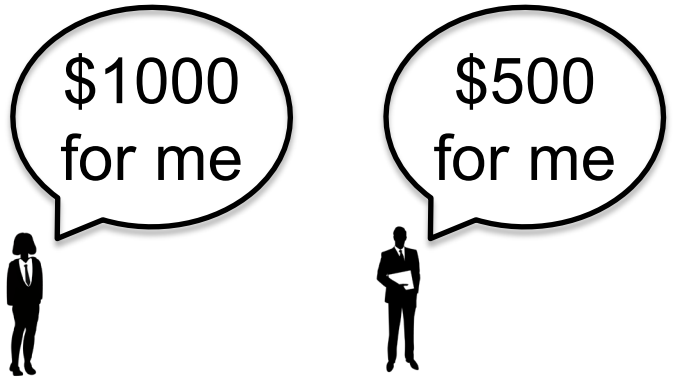


[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

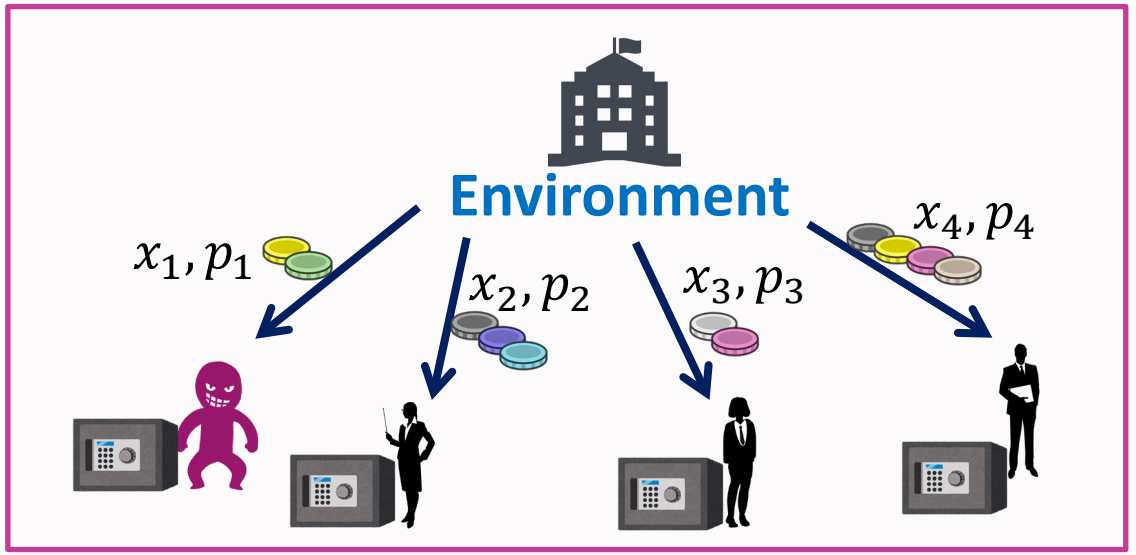
Model Overview

- Security with color coins
  - p_i: price function that determines a price of each color ($\text{colors} \to \text{positive integers}$)
  - Environment distributes input values, color coins, and **price functions** at the onset
    - Price function is for each party

Allow prices to depend on parties



[This work] Security with Color Coins



A Question that Arises Here

Can we realize a secure protocol using color coins?

- Present a positive answer by realizing **secure computation with penalties**
  - Almost the same construction as [BK14]'s protocol (based on indistinguishable coins)
  - Modify the ideal functionality of secure computation with penalties
  - (Assume a common price function for all parties)

- Secure computation with penalties
  - Achieve fairness by imposing financial penalties on adversarial parties
    - No honest party loses coins
    - If an adversary learns the output, but an honest party cannot, then the honest party is compensated with coins

A Question that Arises Here

Can we realize a secure protocol using color coins?

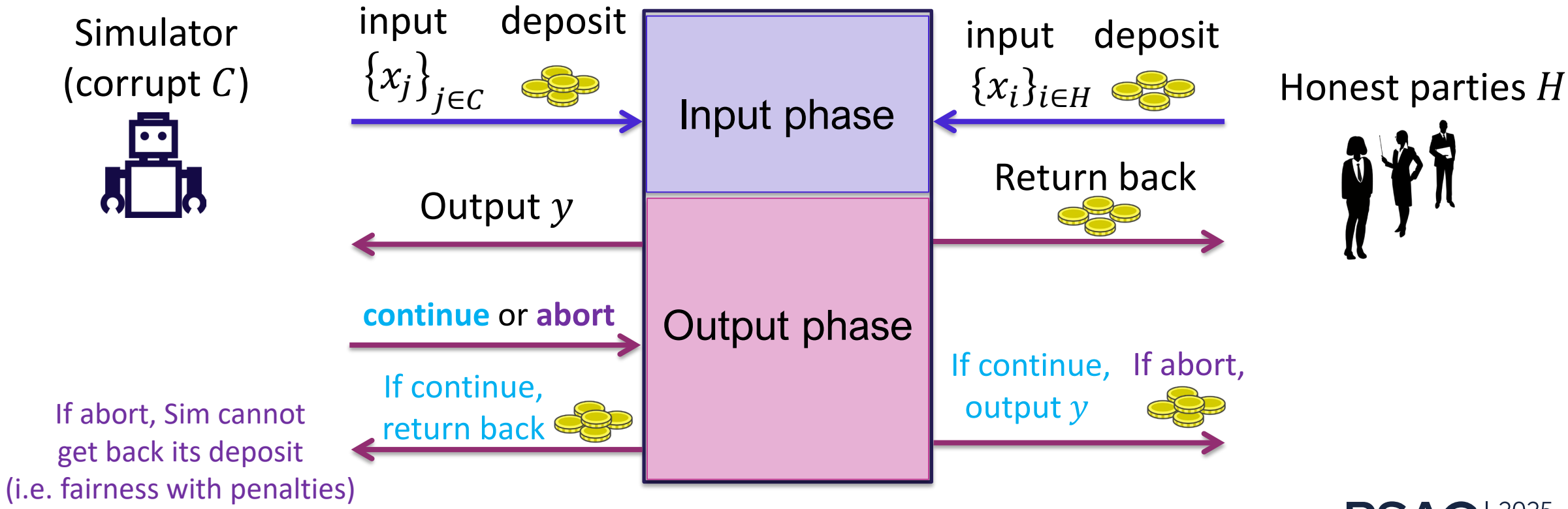
- Present a positive answer by realizing **secure computation with penalties**
  - Almost the same construction as [BK14]'s protocol (based on indistinguishable coins)
  - Modify the ideal functionality of secure computation with penalties
  - (Assume a common price function for all parties)

Center on this topic in the remaining talk

- Secure computation with penalties
  - Achieve fairness by imposing financial penalties on adversarial parties
    - No honest party loses coins
    - If an adversary learns the output, but an honest party cannot, then the honest party is compensated with coins

Secure Computation with Penalties [BK14]

- Overview of the ideal functionality (based on indistinguishable coins)

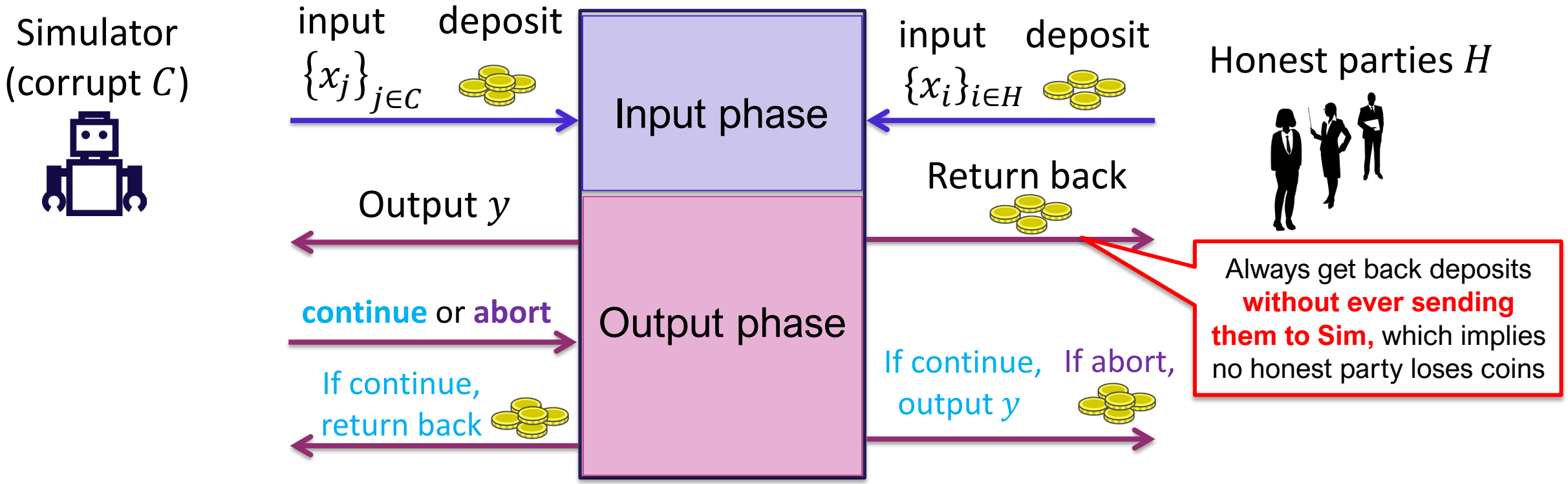


[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

RSAC 2025 Conference

Secure Computation with Penalties [BK14]

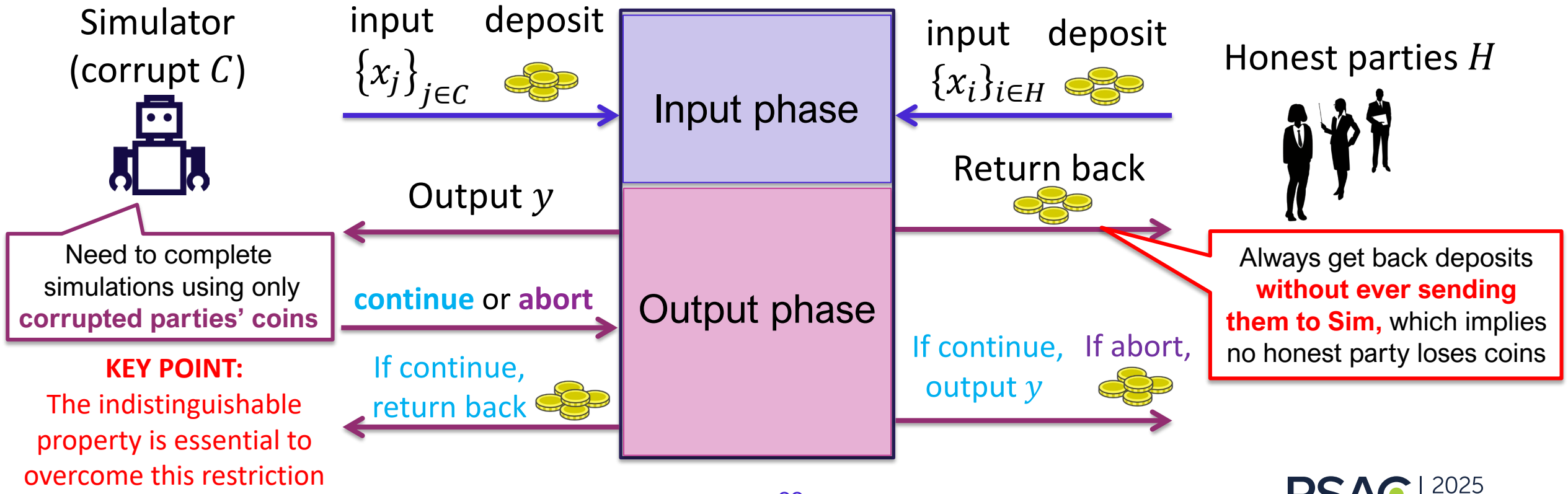
- Overview of the ideal functionality (based on indistinguishable coins)



[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

Secure Computation with Penalties [BK14]

- Overview of the ideal functionality (based on indistinguishable coins)

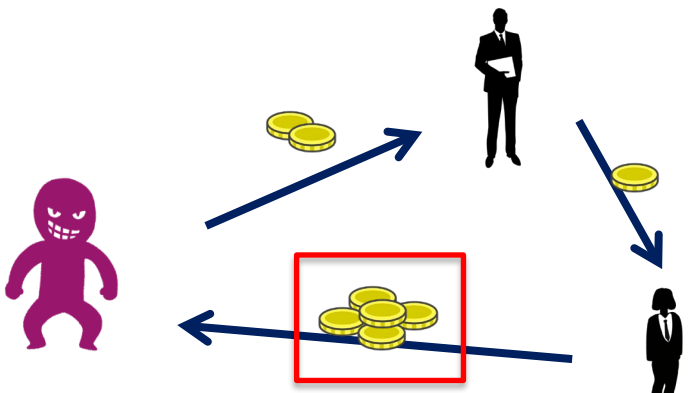


[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

RSAC 2025 Conference

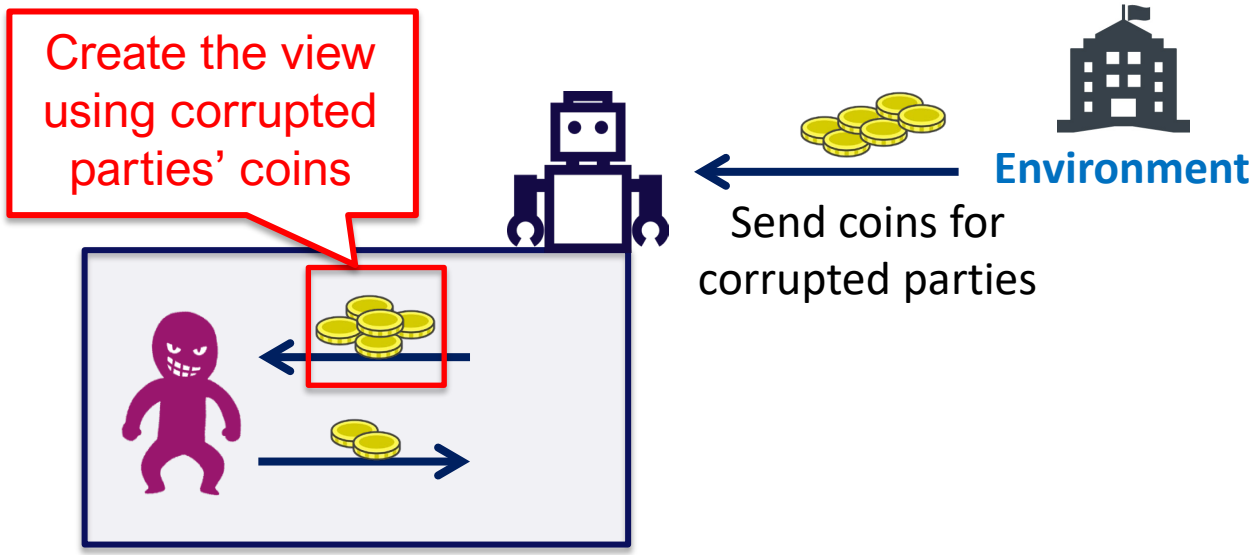
Security Proof with Indistinguishable Coins [BK14]

Real World



Adversary's view contains financial transfers from honest parties

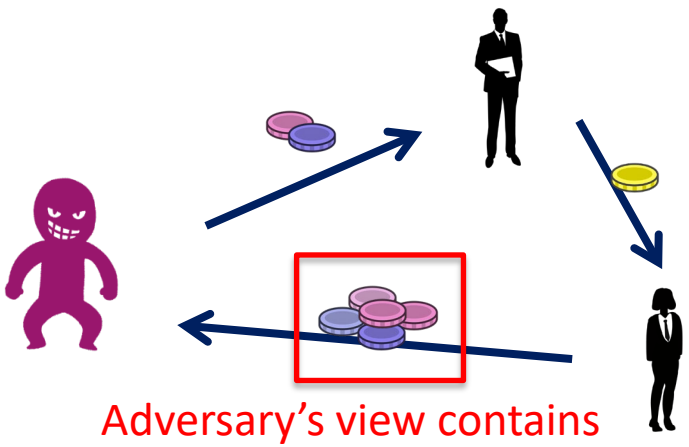
Ideal World



[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

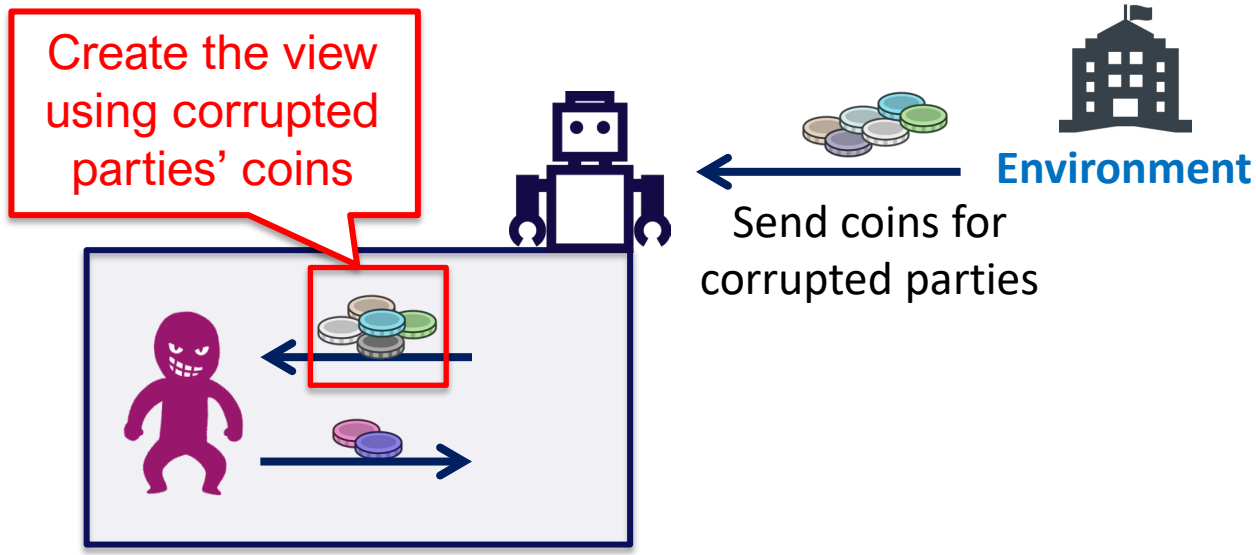
Security Proof with Indistinguishable Coins [BK14]

Real World



Adversary's view contains financial transfers from honest parties

Ideal World



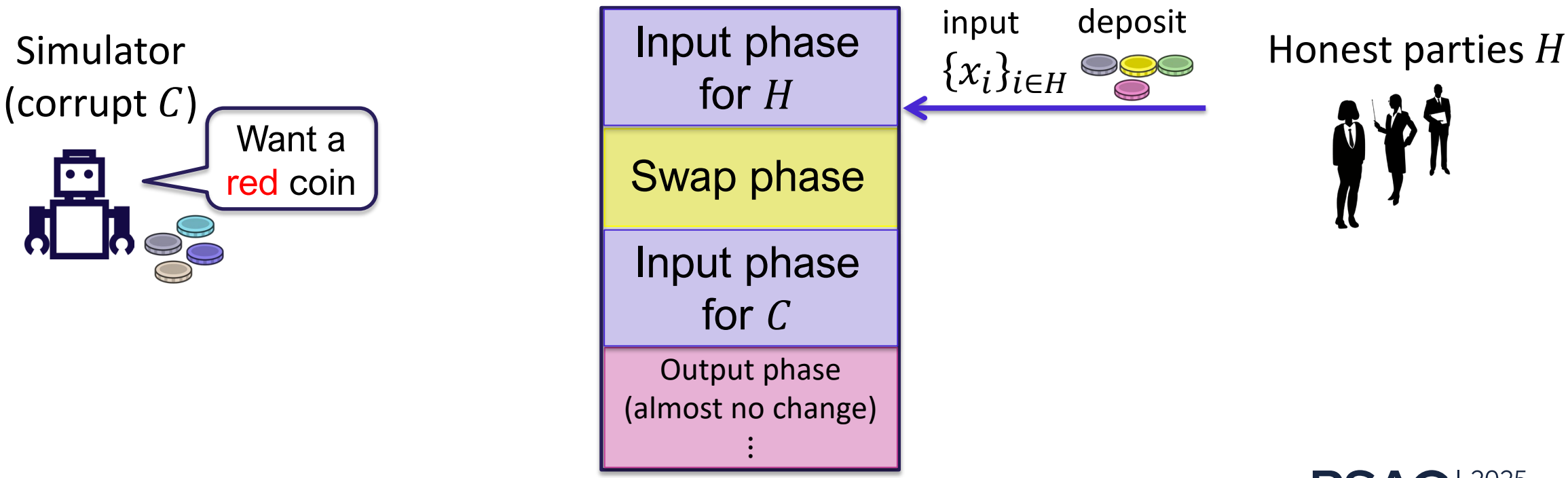
This simulation way does NOT work well if the adversary can distinguish between corrupted parties' coins and honest parties' ones

A workaround needs for security proof based on color coins.

[BK14] I. Bentov and R. Kumaresan, "How to Use Bitcoin to Design Fair Protocols," CRYPTO2014.

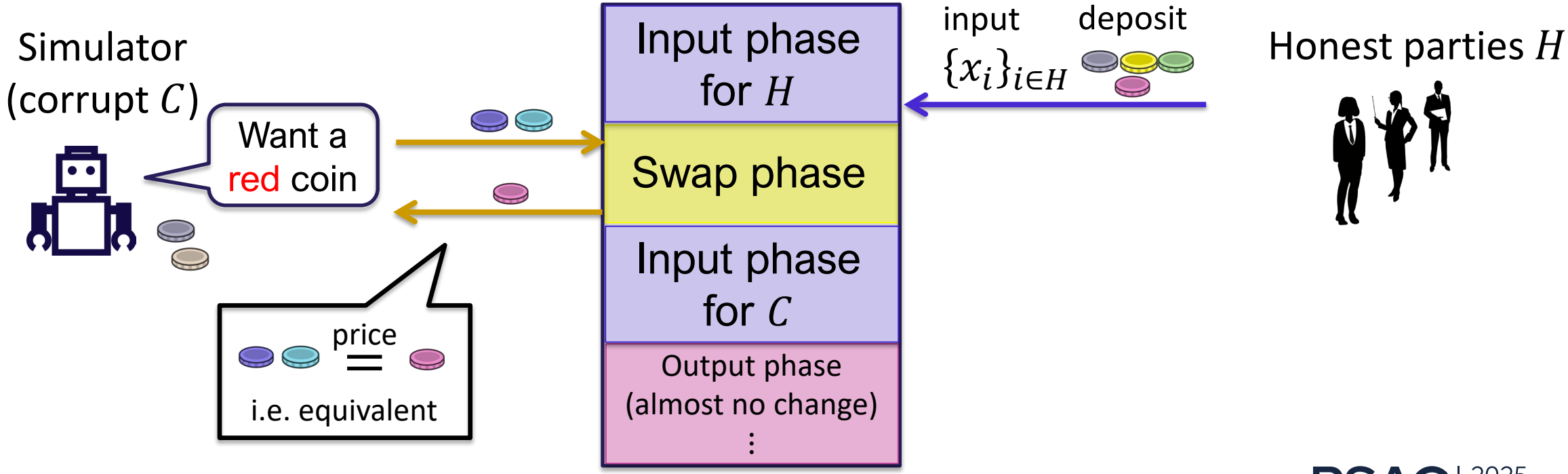
Secure Computation with Penalties (This Work ver.)

- Need a **workaround** for security proof without the indistinguishable property
- Workaround: simulator can **swap** corrupted parties' color coins for honest parties' ones
  - The swap must be **equivalent**



Secure Computation with Penalties (This Work ver.)

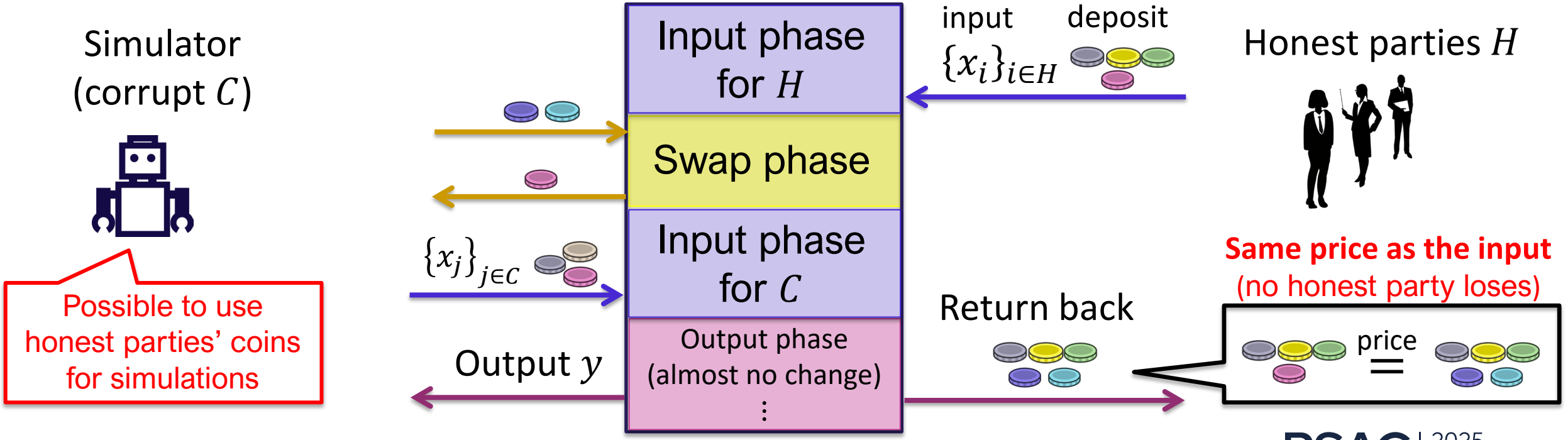
- Need a **workaround** for security proof without the indistinguishable property
- Workaround: simulator can **swap** corrupted parties' color coins for honest parties' ones
  - The swap must be **equivalent**



RSAC 2025 Conference

Secure Computation with Penalties (This Work ver.)

- Need a **workaround** for security proof without the indistinguishable property
- Workaround: simulator can **swap** corrupted parties' color coins for honest parties' ones
  - The swap must be **equivalent**



RSAC 2025 Conference

Summary

- Summary of This Work
  - Propose a new framework: security with color coins
    - Each coin has an identifiable color (i.e. distinguishable from each other)
    - Prices are determined for each color with price functions (i.e. mutually distinct prices)
  - Realize a protocol for secure computation with penalties based on color coins
    - Almost the same procedure as an existing protocol based on indistinguishable coins
    - Security proof without relying on the indistinguishable property of coins
- Future Direction
  - Construct protocols, other than secure computation with penalties, based on color coins
    - e.g., secure lottery, secure cash distribution
  - Construct a protocol that treats price functions as private information

RSAC™ 2025 Conference

Many Voices.
One Community.

Thank you for your kind attention

RSAC 2025 Conference

Many Voices.
One Community.

SESSION ID: CRYP-W08B

Universal Adaptor Signatures from Blackbox Multi-Party Computation

Michele Ciampi, Xiangyu Liu, Ioannis Tzannetos, Vassilis Zikas

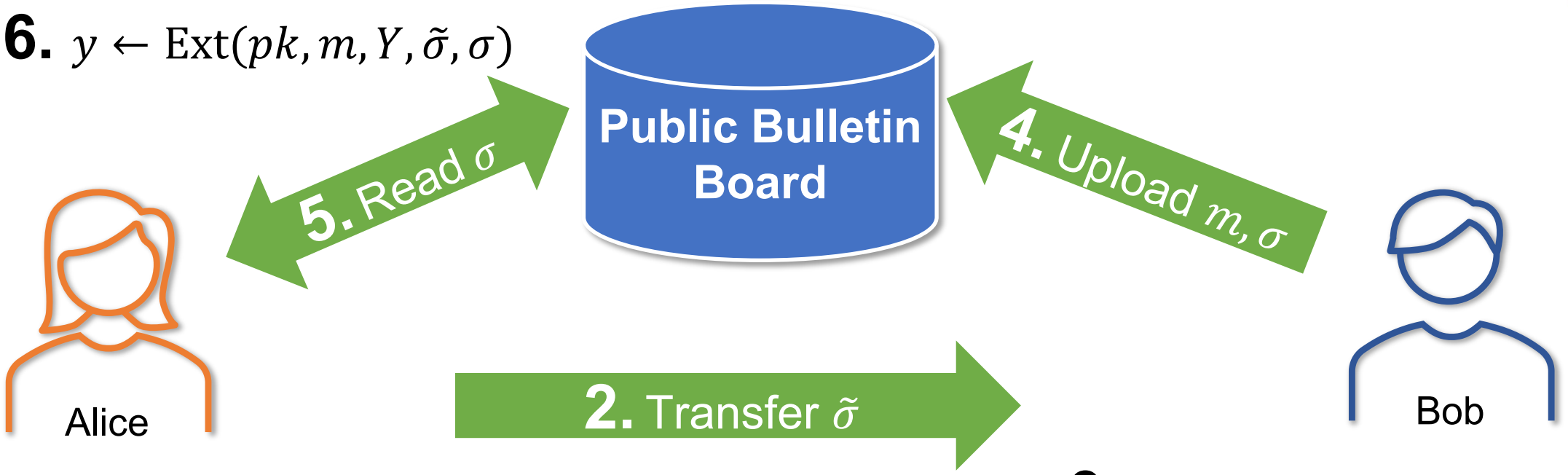
Dr. Xiangyu Liu
University of Edinburgh & Purdue University & NTUA & Georgia Tech

(Adaptor) Signatures

- Signature scheme:
  - key generation: $(pk, sk) \leftarrow \mathrm{Gen}(1^{\lambda})$
  - sign: $\sigma \leftarrow \mathrm{Sign}(sk, m)$
  - verification: $0/1 \leftarrow \mathrm{Ver}(pk, m, \sigma)$
- Adaptor Signature (AS) scheme w.r.t. hard relation R
  - pre-sign: $\tilde{\sigma} \leftarrow \mathrm{pSign}(sk, m, Y)$
  - pre-verification: $0/1 \leftarrow \mathrm{pVer}(pk, m, Y, \tilde{\sigma})$
  - adaption: $\sigma \leftarrow \mathrm{Adapt}(pk, m, \tilde{\sigma}, y)$
  - extraction: $y/\perp \leftarrow \mathrm{Ext}(pk, m, Y, \tilde{\sigma}, \sigma)$

$(Y, y) \in R$
Y an instance (statement)
y a witness of Y

AS for Fair Exchange



1. $\tilde{\sigma} \leftarrow \mathrm{pSign}(sk, m, Y)$

Alice knows (pk, sk) and statement Y
and wants the witness y

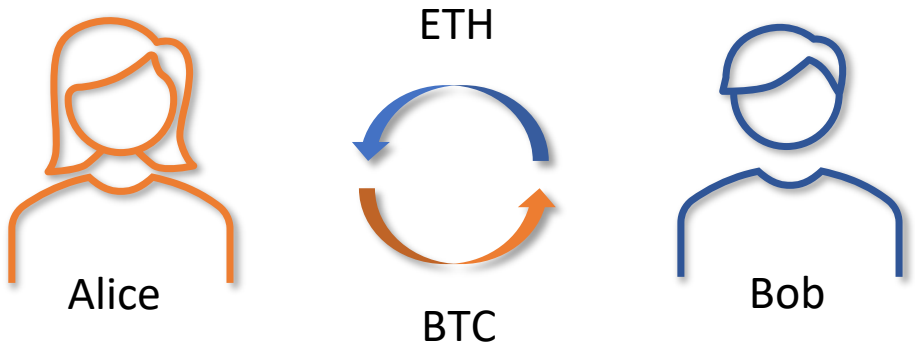
3. $\sigma \leftarrow \mathrm{Adapt}(pk, m, \tilde{\sigma}, y)$

Bob knows (Y, y)
and wants a signature (from Alice) on m

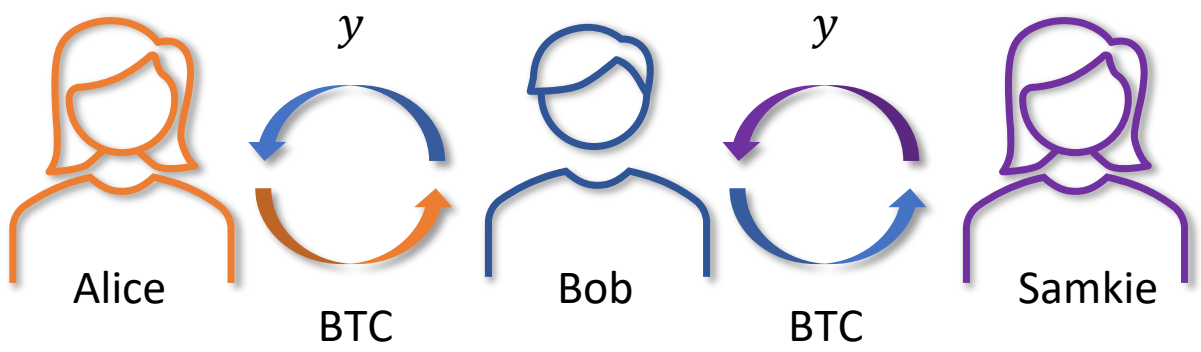
m: transfer coins from Alice to Bob
y: a serial number of some digital service

More Applications of AS

✓ Atomic swap

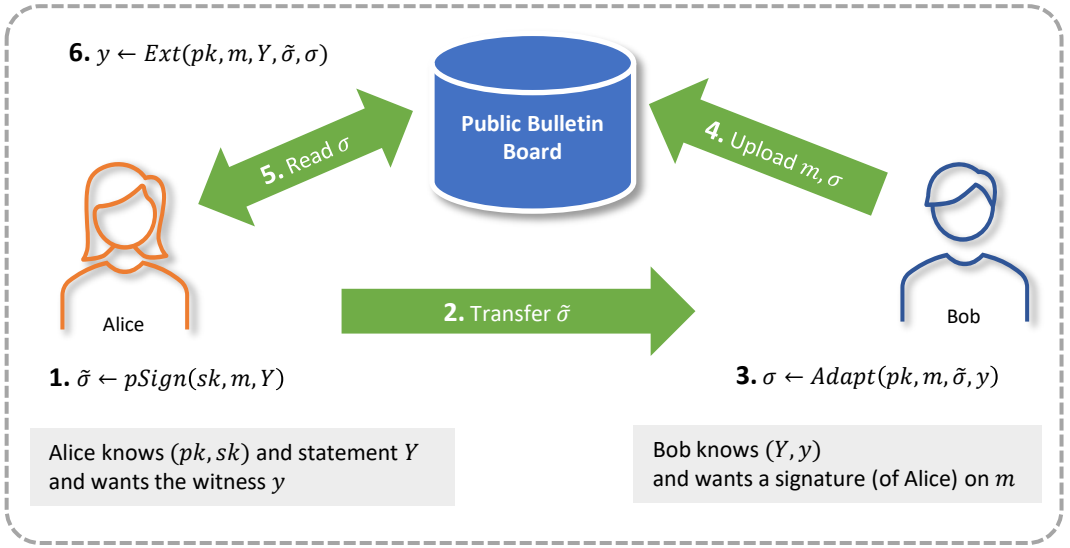


✓ Multi-hop payments



Security of AS

- **1. Unforgeability**
- **2. Witness extractability (sender):**
  - Sender can extract a witness from the valid pre-sig and adapted signature
- **3. Pre-signature adaptability (receiver):**
  - Receiver can adapt a valid pre-sig into a full signature with witness y
- **4. Witness hiding [LTZ24]:**
  - The adapted signature itself does not leak any information about y


6. $y \leftarrow Ext(pk, m, Y, \tilde{\sigma}, \sigma)$
Public Bulletin Board
5. Read σ
4. Upload m, σ
Alice
Bob
2. Transfer $\tilde{\sigma}$
1. $\tilde{\sigma} \leftarrow pSign(sk, m, Y)$
3. $\sigma \leftarrow Adapt(pk, m, \tilde{\sigma}, y)$
Alice knows (pk, sk) and statement Y and wants the witness y
Bob knows (Y, y) and wants a signature (of Alice) on m

Related Works

- Concrete AS schemes
  - ECDSA-based AS [AEE+21 (ASIACRYPT)]
  - Schnorr-based AS [AEE+21 (ASIACRYPT), TZC22 (ISC)]
  - LWE/SIS-based scheme LAS [EEE20 (ESORICS)]
  - Code-based scheme AS [KH22 (Cryptography)]
  - Isogeny-based scheme IAS [TMM21 (FC)]
  - Identification-based AS [EFH+21 (PKC)]
  - AS from CL, BBS+, Waters signatures [GSST24 (EUROCRYPT)]

for a specific Sig scheme and a corresponding language

- Universal Adaptor Signature (UAS) schemes for any NP language
  - GAS1 [DOY22 (INDOCRYPT)] ------no witness hiding!
  - Hamiltonian Cycle-based scheme [LTZ24 (ASIACRYPT)]

[LTZ24]'s Scheme

- pre-signature

$\tilde{\sigma} = (\bar{\sigma}, Y, (cmt, ch = m_0, rsp))$ s.t

- $\bar{\sigma} \leftarrow \mathrm{SIG.Sign}(sk, (m, Y, cmt))$
- m_0 denotes all-zero string
- (cmt, ch, rsp) is a transcript of Sigma protocol for proving y

- full signature

$\sigma = (\bar{\sigma}, Y, (cmt, m \neq m_0, rsp'))$

The scheme needs that:

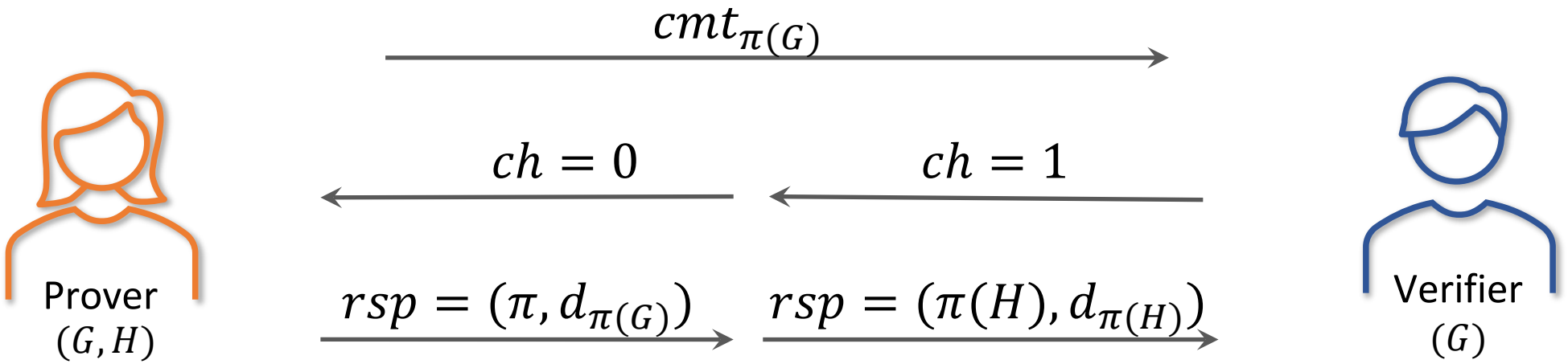
✓ the commitment is not related to the witness

✓ given the commitment for dummy m_0 and witness, one can open this commitment to any other challenge

Does such Sigma protocol exist?

---Yes! Blum's Sigma Protocol for Hamiltonian Cycle!

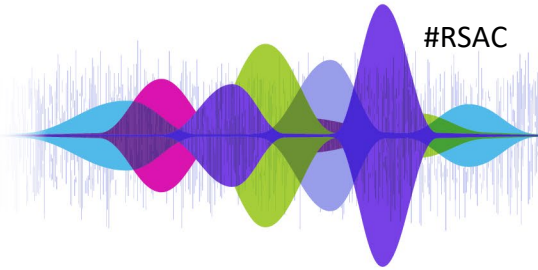
Blum's Sigma Protocol for Hamiltonian Cycle



d: the opening of a commitment

- The commitment is not related to the witness
- With H and $(cmt, ch = 0, rsp)$, cmt can be opened to another challenge $ch = 1$

[LTZ24]'s Scheme

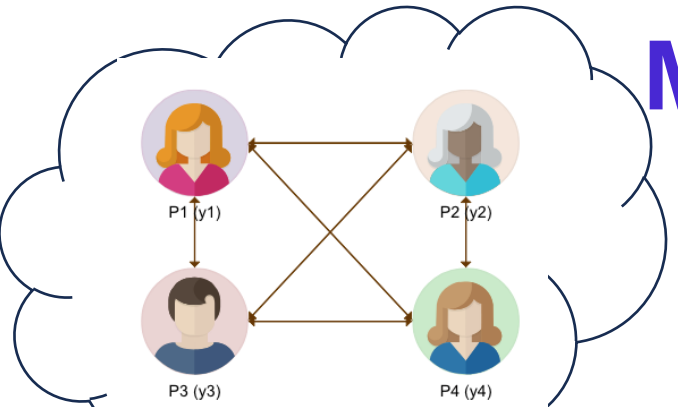


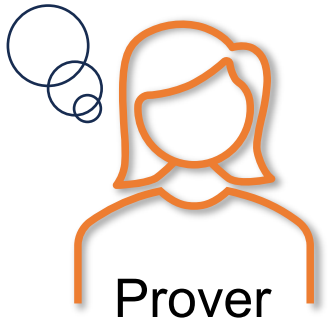
- ✓ the Hamilton cycle problem is NP-complete
- ✓ Due to Karp reduction, any NP relation R can be transferred into a Sigma protocol
- ✓ Therefore, UAS for all NP languages exist (if OWF exist)

Nice, but…

Karp reduction is too heavy. Can we do better? Like, MPC in the head?

MPC-in-the-head [IKOS07]





Prover
(Y, y)

Commit $V_1, \ldots, V_N$

Random i

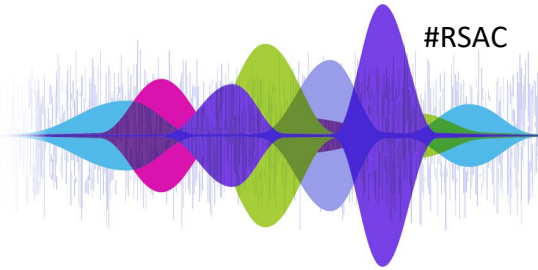
Open $\{V_1, \ldots, V_N\} \setminus V_i$

Verifier
(Y)

Prover imagines that N parties execute an MPC protocol Π for

$$f(Y, y_1, y_2, \ldots, y_N) = 1 \text{ iff } \left(Y, \bigoplus_{i \in [N]} y_i\right) \in R$$

Problem of Using MPCitH



- In MPC-in-the-head, the commitment depends on the witness.
- However, in pre-signing, only statement Y is known but not the witness.

[LTZ24]'s scheme:

- pre-signature: $\tilde{\sigma} = (\bar{\sigma}, Y, (cmt, ch = m_{\mathbf{0}}, rsp))$
- full signature: $\sigma = (\bar{\sigma}, Y, (cmt, m \neq m_{\mathbf{0}}, rsp'))$

Our Solution

[LTZ24]'s scheme:

- pre-signature: $\tilde{\sigma} = (\bar{\sigma}, Y, (cmt, ch = m_{\mathbf{0}}, rsp))$
- full signature: $\sigma = (\bar{\sigma}, Y, (cmt, m \neq m_{\mathbf{0}}, rsp'))$

Our solution:

- Pre-signature: $\tilde{\sigma} = (\bar{\sigma}, Y, (pk^e, sk^e))$
- Full signature: $\sigma = (\bar{\sigma}, Y, pk^e, \pi)$
- ---π is the zero-knowledge proof based on MPCitH with pk^e the commitment key

$\mathrm{Commit}(m)$: $c = \mathrm{PKE.Enc}(pk^e, m; r)$

$\mathrm{Open}(c)$: Return (m, r) and check if $c = \mathrm{PKE.Enc}(pk^e, m; r)$

- [LTZ24] uses a standard commitment scheme
- If we implement the commitment with a PKE scheme, then sk^e can be used to decrypt the witness
- The adaptor now uses pk^e as the commitment key to calculate transcripts of the MPC-in-the-head paradigm

Some Subtleties

- Pre-signature: $\tilde{\sigma} = (\bar{\sigma}, Y, (pk^e, sk^e))$
- Full signature: $\sigma = (\bar{\sigma}, Y, pk^e, \pi)$

- What if sk^e fails to extract y from the commitments?
  - we require prefect correctness of PKE
  - **not enough, since (pk^e, sk^e) maybe ill-formed**
  - pre-signature includes randomness r in key generation instead of (pk^e, sk^e)
- What if a malicious signer takes some "BAD" key pair s.t. pk^e will fully expose the View?
  - does not conflict to the witness hiding (signature adapted from a normally generated $\tilde{\sigma}$ does not leak y)
  - a malicious signer can always extract y and expose it

Our Scheme

- pre-signature

$$\tilde{\sigma} = (\bar{\sigma}, r) \quad \text{s.t.} \quad \begin{cases} (pk^e, sk^e) \leftarrow \mathrm{PKE.Kgen}(1^\lambda; r) \\ \bar{\sigma} \leftarrow \mathrm{SIG.Sign}\big(sk, (m, Y, pk^e)\big) \end{cases}$$

- full-signature

$$\sigma = (\bar{\sigma}, Y, pk^e, \pi))$$

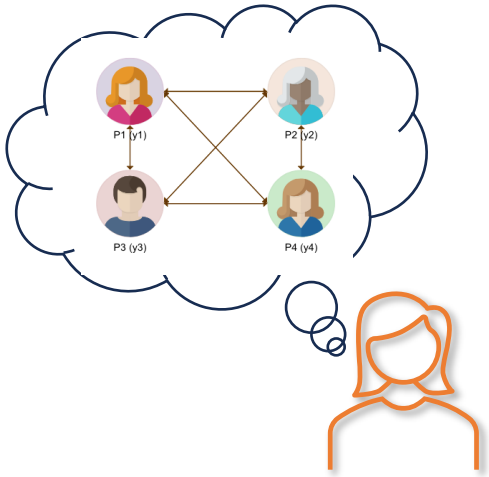
π is the zero-knowledge proof based on MPCitH using pk^e as the commitment key

- Extract

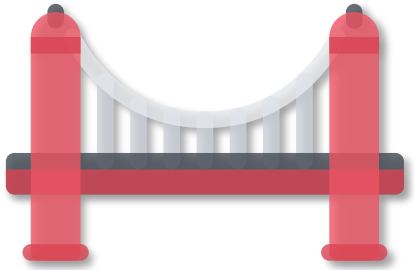
recompute sk^e and extract y from π

Apply & Conclusion

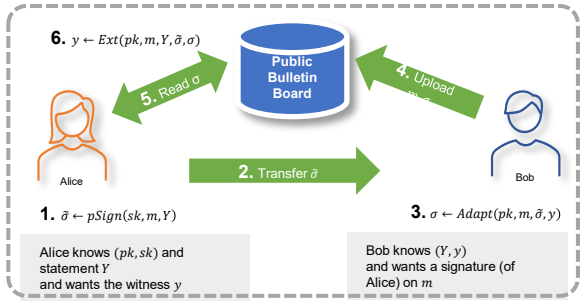
- Efficient universal adaptor signatures (UAS) from MPCitH
- Enhance the applicability of decentralized applications from AS



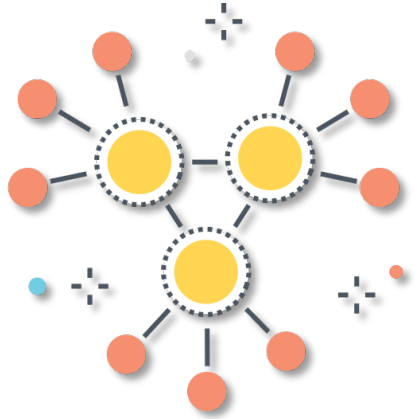
MPC



Our work


6. $y \leftarrow Ext(pk, m, Y, \tilde{\sigma}, \sigma)$
Public Bulletin Board
5. Read σ
4. Upload
Alice
2. Transfer $\tilde{\sigma}$
Bob
1. $\tilde{\sigma} \leftarrow pSign(sk, m, Y)$
3. $\sigma \leftarrow Adapt(pk, m, \tilde{\sigma}, y)$
Alice knows (pk, sk) and statement Y and wants the witness y
Bob knows (Y, y) and wants a signature (of Alice) on m

efficient UAS



Decentralized App

References

- [TZC22] Tu, B., Zhang, M., & Yu, C. (2022). *Efficient ECDSA-based adaptor signature for batched atomic swaps.* In **ISC 2022**.
- [EEE20] Esgin, M.F., Ersoy, O., & Erkin, Z. (2020). *Post-quantum adaptor signatures and payment channel networks.* In **ESORICS 2020**.
- [TMM21] Tairi, E., Moreno-Sanchez, P., & Maffei, M. (2021). *Post-quantum adaptor signature for privacy-preserving off-chain payments.* In **FC 2021**.
- [KH22] Klamti, J.B., & Hasan, M.A. (2022). *Post-quantum two-party adaptor signature based on coding theory.***Cryptography, 6**(1), 6.
- [EFH+21] Erwig, A., Faust, S., Hostáková, K., Maitra, M., & Riahi, S. (2021). *Two-party adaptor signatures from identification schemes.* In **PKC 2021**.
- [DOY22] Dai, W., Okamoto, T., & Yamamoto, G. (2022). *Stronger security and generic constructions for adaptor signatures.* In **INDOCRYPT 2022**.
- [GSST24] Gerhart, P., Schröder, D., Soni, P., & Thyagarajan, S.A.K. (2024). *Foundations of adaptor signatures.* In **EUROCRYPT 2024**.
- [LTZ24] Liu, X., Tzannetos, I., & Zikas, V. (2024). *Adaptor signatures: New security definition and a generic construction for NP relations.* In **ASIACRYPT 2024**.
- [IKOS07] Ishai, Y., Kushilevitz, E., Ostrovsky, R., & Sahai, A. (2007). *Zero-knowledge from secure multiparty computation*. In **STOC 2007**.

RSAC™ 2025 Conference

Many Voices.
One Community.

Thank You!

For more information:

https://eprint.iacr.org/2024/1773